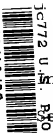


04/12/00



35772 U.S. Pat.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

re Application of Jeff Wasilko

Serial No.: Not yet assigned

Group Art Unit:

Filed: Herewith

Examiner:

For: ATTENUATION, DELAY, QUEUING, AND MESSAGE CACHEING PROCESSES  
FOR USE IN E-MAIL PROTOCOLS IN ORDER TO REDUCE NETWORK SERVER  
LOADING

Assistant Commissioner for Patents  
Box PATENT APPLICATION  
Washington, D.C. 20231

Dear Sir:

Enclosed please find the following:

1. Specification, abstract and claims (3 independent, 23 dependent, 26 total) (29 pages);
2. Informal drawings (8 figures, 6 sheets);
3. Declaration and Power of Attorney;
4. Assignment with Recordation Sheet;
5. One check in the amount of \$838.00 (\$690.00 for filing the application, \$40.00 for filing the assignment, and \$108.00 for filing excess claims); and,
6. Certificate of Express mailing.

The Commissioner is hereby authorized to charge any fee deficiency, or credit any overpayment, to Deposit Account No. 18-1579. The Commissioner is also authorized to charge Deposit Account No. 18-1579 for any future fees connected in any way to this application. Two copies of this letter are enclosed.

Respectfully submitted,

Kevin L. Pontius  
Registration No. 37,512  
Roberts Abokhair & Mardula, LLC  
11800 Sunrise Valley Drive, Suite 1000  
Reston, VA 20191-5302  
(703) 391-2900

04/12/00  
80395/60  
PLD S N 0891



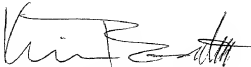
04/12/00 04/12/00

CERTIFICATE OF EXPRESS MAILING

Express Mail Mailing Label Number EL528296366US

Date of Deposit: April 12, 2000

I hereby certify that the patent application of Jeff Wasilko for an **ATTENUATION, DELAY, QUEUING, AND MESSAGE CACHEING PROCESSES FOR USE IN E-MAIL PROTOCOLS IN ORDER TO REDUCE NETWORK SERVER LOADING** including the specification, abstract, and claims (3 independent, 23 dependent, 26 total)(29 pages); informal drawings (8 figures, 6 sheets); declaration and power of attorney; an assignment together with a recordation cover sheet; and a check in the amount of \$838.00 (\$690.00 for filing the application, \$40.00 for filing the assignment, and \$108.00 for filing excess claims), are being deposited with the United States Postal Service for "Express Mail" service under 37 C.F.R. § 1.10 on the date indicated above and are addressed to the Assistant Commissioner for Patents, Box Patent Application, Washington, D.C. 20231.



Kevin L. Pontius  
Registration No. 37,512  
Roberts Abokhair & Mardula, LLC  
11800 Sunrise Valley Drive, Suite 1000  
Reston, VA 20191-5302  
(703) 391-2900

April 12, 2000

Atty. Docket No. 2585-003

1 Title: **ATTENUATION, DELAY, QUEUING, AND MESSAGE CACHEING**  
2 **PROCESSES FOR USE IN E-MAIL PROTOCOLS IN ORDER TO**  
3 **REDUCE NETWORK SERVER LOADING**

4 Inventor: **Jeff Wasilko**  
5

6 **RELATED APPLICATION**

7 This application is related to the commonly-assigned, co-pending patent  
8 application of Howard Pfeffer and John Leddy entitled Reduction of Network Server  
9 Loading, filed March 23, 2000, which has been assigned serial no. 09/533,463, and which  
10 is incorporated herein by reference in its entirety for all purposes.

11 **BACKGROUND OF THE INVENTION**

12 **1. Field of the Invention**

13 The present invention relates generally to the field of data networks. More  
14 particularly, the present invention relates to reduction of traffic handling load on network  
15 servers by decentralization of mail handling protocols.

16 **2. Background Information**

17 Over the last twenty years, the demand for data network services has grown  
18 rapidly. Many large networks have been built by a number of providers to meet the  
19 voracious demand for bandwidth to handle data traffic.

20 Data networks are commonly used for *inter alia* transmission of electronic mail  
21 messages (hereinafter e-mail). According to Post Office Protocol (hereinafter POP)  
22 standard for e-mail handling, transmitted e-mail messages are routed to a centralized mail  
23 server facility. Those e-mail messages are warehoused at the centralized mail server until  
24 retrieved by their intended recipients. A user retrieves their e-mail messages from the mail

1 server by sending an inquiry message via the network to the mail server asking if there is  
2 any mail stored there for them. Commonly these inquiry messages sent to the mail server  
3 are known as "POP checks" because they check for mail according to the Post Office  
4 Protocol. If the user's mail box is empty, then the mail server sends a negative response to  
5 the user telling him so. On the other hand, if the mail server is storing mail messages for  
6 the user, those messages are transmitted (in response to the POP check) to the user via the  
7 network.

8 One problem with large data networks where the clients are connected at all times  
9 without having to create a dial up connection is the large amount of network traffic due to  
10 frequent POP checks that users make (or, more typically, that the users' computers makes  
11 on the users' behalf) to see if there are any new e-mail messages waiting for them on the  
12 mail server. The use of a POP3 mail system in a wide area network (WAN) may result in  
13 a large amount of network traffic. This is not only a bandwidth problem, but also causes a  
14 substantial loading on the servers across the network that have to handle and route all this  
15 largely unproductive traffic. When a mail server is remote from the mail client, each  
16 POP3 request may require numerous hops to transverse the network, and response must  
17 travel the same distance.

18 It is largely unproductive traffic because the vast bulk of POP checks (over 90%,  
19 typically) result in negative responses because POP checks are generated much more  
20 frequently than the frequency with which e-mail messages arrive at the mail server. This  
21 is very inefficient. Significant traffic handling server load reduction, and some bandwidth  
22 savings, can be made if most POP checks are terminated close in the network to the  
23 sender.

1           Thus, what is needed is a scheme for reducing the number of POP checks that are  
2 transmitted over the network to the mail server.

3           Another e-mail related problem in networks is that the bandwidth demand resulting  
4 from e-mail traffic is concentrated during certain times of the day. In particular, the  
5 morning hours are a concentrated time for retrieval of e-mail messages from the mail  
6 server. These e-mail-generated spikes in bandwidth demand present network management  
7 challenges. One typical solution is to increase bandwidth capacity of the network to  
8 accommodate demand spikes completely. Obviously, this is an expensive and inefficient  
9 option because the added capacity will go largely unused (except in the event of demand  
10 spikes). Another typical solution is to simply permit poor network performance during  
11 periods when bandwidth demand spikes. Obviously, this option would be a source of  
12 irritation to the users of the network.

13           Thus, what is needed is a scheme to accommodate e-mail retrieval traffic without  
14 adding capacity that will go largely unused and without reducing network performance.

#### 15                               **SUMMARY OF THE INVENTION**

16           It is an object of the present invention to a scheme for reducing the number of POP  
17 checks that are transmitted over the network to a POP mail server.

18           It is another object of the present invention to scheme to accommodate e-mail  
19 retrieval traffic without adding capacity that will go largely unused and without reducing  
20 network performance.

21           It is yet another object of the present invention to efficiently distribute e-mail  
22 handled according to a POP system.

23           It is still another object of the present invention to push bandwidth demand and  
24 traffic handling load toward the edges of a wide area network.

CONFIDENTIAL

1           The present invention addresses this problem with three approaches that may be  
2       used separately or together. The first is to attenuate the POP checks. The second is to  
3       delay the POP checks. The third is to cache the e-mail messages.

4           Attenuation of POP checks is accomplished by intercepting each POP check packet  
5       at a proxy server that is nominally local to where the user is located. The proxy server lets  
6       the user's first POP check proceed on through the network to the mail server. Thereafter,  
7       though, the proxy server only permits that user's received POP checks to proceed onward  
8       according to a predetermined algorithm. For example, the proxy server may only permit a  
9       POP check to proceed to the mail server if it has been at least fifteen minutes since the last  
10      time the mail server was actually checked for e-mail by that particular user. When overly  
11      frequent POP checks by that user are received prior to the permitted time, no actual check  
12      of the mail server is permitted and the proxy server simply informs the user that he has no  
13      mail (despite not knowing deterministically whether that is a true statement).

14          As with attenuation, delay of POP checks is accomplished by intercepting each  
15      POP check packet at a proxy server that is nominally local to where the user is located.  
16      Although an initial POP check is permitted to proceed on through the network to the mail  
17      server, if the frequency of subsequent POP checks is greater than a threshold, they are  
18      placed in a delay queue and processed with a delay inserted into the signaling. This delay  
19      can be a fixed value or a variable that changes with client POP habits. This delay can be  
20      inserted for a single POP command or for multiple of the POP session commands  
21      depending on the total amount of delay desired. The effect of this delay is to increase the  
22      time that it takes for a POP session to complete. An advantage of this technique is that the  
23      state of the mailbox is accurately and deterministically reported.

1 According to the cache aspect of the invention, a user's e-mail is cached at the  
2 proxy server nearest to his presumed location. This decentralizes the e-mail storage away  
3 from the mail server and spreads it out over the network at the various proxy servers. This  
4 cache action is preferably done when there is a lull in network traffic (e.g., at night). This  
5 can also be done as soon as mail is available so that the proxy deterministically knows that  
6 a client has unread mail. This also has the effect of decentralizing the bandwidth demand  
7 on the overall network since the e-mail messages have a shorter distance to travel when  
8 retrieved by the user from the cache location at the proxy server.

#### 9 BRIEF DESCRIPTION OF THE DRAWINGS

10 Additional objects and advantages of the present invention will be apparent in the  
11 following detailed description read in conjunction with the accompanying drawing figures.

12 **Fig. 1** illustrates a network diagram implementing proxy handling of e-mail  
13 according to an embodiment of the present invention.

14 **Fig. 2** illustrates a logical implementation for a subset of the signaling transactions  
15 that take place between a client, a proxy server, and a mail server according to an  
16 embodiment of the present invention.

17 **Fig. 3** illustrates a logical implementation for another subset of the signaling  
18 transactions that take place between a client, a proxy server, and a mail server according to  
19 an embodiment of the present invention.

20 **Fig. 4** illustrates a logical implementation for a subset of the signaling transactions  
21 that take place between a client, a proxy server, and a mail server according to an alternate  
22 embodiment of the present invention.

1           **Fig. 5** illustrates a logical implementation for a subset of the signaling transactions  
2 that take place between a client, a proxy server, and a mail server according to a further  
3 alternate embodiment of the present invention.

4           **Fig. 6** illustrates a logical implementation for a subset of the signaling transactions  
5 that take place between a client, a proxy server, and a mail server according to another  
6 alternate embodiment of the present invention.

7           **Fig. 7** illustrates a logical implementation for a subset of the signaling transactions  
8 that take place between a client, a proxy server, and a mail server according to still another  
9 alternate embodiment of the present invention.

10          **Fig. 8** illustrates a logical implementation for a subset of the signaling transactions  
11 that take place between a client, a proxy server, and a mail server according to yet another  
12 alternate embodiment of the present invention.

#### 13                   **DETAILED DESCRIPTION OF THE INVENTION**

14          Typical e-mail client programs support a feature that performs automated mailbox  
15 checking at configurable intervals. A user may set their e-mail program to query for new  
16 mail every five minutes, for example. The "always on" characteristic of a broadband  
17 Internet service enables a subscriber to leave their e-mail program running and  
18 continuously polling for mail. This adds further to the load of POP3 packets on the  
19 service provider's network. And, while polling intervals will vary, empirical observation  
20 shows that over 90% of the mailbox queries return no new mail. This problem of empty  
21 POP checks is addressed by the attenuation aspect of the present invention.

22          Attenuation of POP checks is accomplished by a proxy server, which implements a  
23 software proxy. The proxy server intercepts POP requests that originate with its locally  
24 situated network component. As a proxy, it responds to a majority of these POP checks



1 based on its most recent knowledge of the state of the requestor's actual mailbox. The  
2 proxy server uses an "attenuation interval" of  $n$  minutes. Thus, every  $n$  minutes it will  
3 allow a mail query session (i.e., POP check) to flow through to the actual mail server. If  
4 the mailbox is empty after this session, the proxy server will note that, and will return  
5 "mailbox empty" responses to all POP checks for that mailbox for the next  $n$  minutes,  
6 after which it will permit the subsequent POP check packet to flow through for that  
7 mailbox, and so continue the pattern.

8 Generally speaking, the proxy server permits a user's first POP check to proceed  
9 on through the network to the mail server. Thereafter, though, the proxy server only  
10 permits that user's received POP checks to proceed onward according to a predetermined  
11 algorithm. For example, the proxy server may only permit a POP check to proceed to the  
12 mail server if it has been at least fifteen minutes since the last time the mail server was  
13 actually checked for e-mail by that particular user. When overly frequent POP checks by  
14 that user are received prior to the permitted time, no actual check of the mail server is  
15 permitted and the proxy server simply informs the user that he has no mail (despite not  
16 knowing deterministically whether that is a true statement).

17 While it is a worthy object to moderate the amount of traffic handling that is  
18 required by the servers in a network, the attenuation solution should be implemented with  
19 respect for the concerns of the e-mail users. Under certain circumstances, an e-mail user  
20 may observe that the attenuation algorithm as described above may cause some delay in  
21 how quickly they receive their e-mail messages. The user may perceive this not as an  
22 optimization but, rather, as poor service. Accordingly, there is a need to accommodate the  
23 expectations of the e-mail users to the extent possible.

One simple way to avoid user perception of delay service of e-mail messages is to set the interval of  $n$  minutes at which POP check flow through is permitted as low as possible. In fact, the value of  $n$  can be set dynamically, based on overall network load, so that it is low when network load is light and set higher when network load peaks.

An optional feature of this attenuation process is that a custom SMTP extension may provide an event service whereby the proxy server can receive notification when new mail arrives for any mail account on the mail server for which it is acting as an attenuation proxy. In case of such an event notification, it will permit the subsequent POP check packet to flow through for that mailbox.

Another optional algorithm is to permit POP checks to flow through based upon the number of POP checks that have been received for a given e-mail account. In other words, only one out of every  $m$  POP checks is permitted to flow through to the mail server. For obvious reasons, this is not a preferred method. It is mentioned simply because of the ease with which it could be implemented.

Another way to manage user expectations is to tell a user that his e-mail service is being attenuated. This is implemented by causing the proxy server to automatically send an e-mail message to affected users on a regular basis (e.g., once a day, or any other choice of programmable period). The contents of the e-mail communicate to the user that their frequent POP checks are placing unproductive load on the system and that the excessive checks will be attenuated. Preferably, message also notifies the recipient that as they reduce the frequency of their POP checks they will no longer be attenuated and directs them to a web page with details on how to set the frequency in their e-mail client software. Additionally, instructions on how to modify email client preferences or a reference to other documentation such as a user guide may be contained in the notification

1 email. The reference to other documentation is preferably embodied as a link to a web  
2 page that provides relevant instructions.

3 The preferred attenuation algorithm is a combination of a time-based rule (i.e., wait  
4  $n$  minutes before letting another POP check through) and a demand-based rule (i.e., let the  
5 next POP check through only if notice of actual mail receipt has been received from the  
6 mail server). Although the demand-based rule alone may appear to be adequate, the  
7 combination with a time-based rule ensures that e-mail still gets checked in the event that  
8 the notification message from the mail server is not sent or fails to be routed to the proxy  
9 server. The redundant use of the two rules provides a more robust system, and minimizes  
10 the chances of causing user dissatisfaction with e-mail service.

11 The attenuation aspect of the present invention preferably makes a deterministic  
12 assessment of whether an e-mail account being attenuated actually has any new e-mail  
13 messages to be retrieved. One method is for the proxy server to snoop the actions of the  
14 POP protocol and keep track of how many messages are read from the mail server and  
15 how many were deleted from the mail server.

16 A simpler method to accomplish this is to wait until the e-mail client ends the e-  
17 mail session. This happens when the e-mail client transmits a "quit" message. Rather  
18 than letting the quit message flow through immediately, the proxy server temporarily  
19 retains the quit message while it determines the status of that e-mail account. While the  
20 quit command is being held, the proxy server sends a "stat" command to the mail server  
21 asking, in essence, "Do you have mail for me?" If the answer received from the mail  
22 server is "no," then the proxy server knows deterministically that the e-mail client has no  
23 un-retrieved messages. This status is cached locally at the proxy server. Subsequently,  
24 the quit message is permitted to travel onward through the network to the mail server.

1           One algorithm for implementing the attenuation aspect of the present invention  
2       uses a set of tables. When a new user checks their e-mail account on the mail server (i.e.,  
3       sends an initial POP check), the proxy server places that user in an attenuation table.  
4       Assuming that the state of the user's mailbox is determined, a timer is started when the  
5       user is placed in the table. The user remains in the table until the timer times out. Each  
6       time the user transmits a POP check, their POP check is attenuated at the proxy server if  
7       the user still remains listed in the table. In other words, until the timer times out and  
8       removes the user from the attenuation table, that user's mail client software will continue  
9       to receive pseudo responses to its POP checks telling the user that the mail server has no  
10      mail for that account.

11           Another triggering event for removing the user from the attenuation table is if a  
12      notification message is received from the e-mail server that the user has mail. By being  
13      removed from the table, the user's next POP check is passed through to the mail server.

14           Every time the user is removed from the attenuation table (for whatever reason:  
15      time out, notification, server reset, etc.) that user's next POP check is permitted to flow  
16      through to the mail server. Concurrently, the transmission of that POP check places the  
17      user back into the attenuation table and re-starts the timer. In this way, the user's own  
18      actions re-establish the state in an attenuation queue.

19           In forming the tables in this implementation, the username information is cached.  
20      To avoid the need to perform linear searches, a hash table algorithm is used. Each hash  
21      entry in the table is a fixed length value, e.g., a five hash, created according one of plural  
22      hash algorithms that are well known in the database art. These hashes are used as keys for  
23      entry in a hash table. The first time a POP check is seen from a user, that user's username  
24      is hashed and entered as an entry in the hash table. The hash table uses the key-value pair.

1 The key is the hash, and the value is the pointer to a small data structure somewhere in the  
2 proxy server's memory. The data structure contains the username in clear text, as well as  
3 the user's password in clear text. The data structure also may contain information  
4 indicating the state of the user's mailbox, and a value indicating the elapsed time left for  
5 that user.

6 This implementation also uses another table, which is an ordered table that  
7 indicates which one of the hash values (indicative of the individual users) will expire next.  
8 The ones that will expire soonest are placed at the top of the ordered table, and the ones  
9 that will expire last are placed at the bottom of the ordered table. The ordered table may  
10 be thought of as a timer list.

11 Thus, when a user first checks for e-mail, their username is hashed and the hash is  
12 placed both in a hash table and at the end of the timer list ordered table. An algorithm  
13 then regularly checks the timer list to see if one or more of the hashes at the top of the list  
14 have timed out.

15 A more detailed discussion of this preferred implementation is discussed as  
16 follows.

17 The attenuation algorithm has two main logical components -- a table of users and  
18 a finite state machine. The finite state machine represents the logic that maps to any  
19 particular combination of user state and POP3 command.

20 The user table is a hash table where the hash key is the user's mailbox account  
21 name for the POP3 server that the proxy server is proxying. A user's entry in the table has  
22 attributes, including mailbox password (may be encrypted or plain text), a time stamp of  
23 the last time the mailbox was verified as empty, state (where the state is the user's current

state in the sequence of commands that make up a POP3 session), lock flag, terminate flag, timeout field, etc.

### States, Events, and Actions

What follows is a list of possible states for a POP3 attenuation session, and the actions and state transitions that result from various events, where the main event type is a POP3 command. A session associates a command with a mailbox username/usertable entry; although the POP3 **user** command is the only command that carries that name, an implementation mechanism can make this association for subsequent commands within the session.

None State. Event: POP3 proxy server receives the **user** command, but there is no corresponding entry in the user table. The POP3 proxy server creates a table entry for this user, forwards the command to the POP3 mail server, and returns the response to the client. The state transitions to Authorization.

Idle State. Event: POP3 proxy server receives the **user** command and finds a corresponding entry in the table. Sends +OK response to the client. The state transitions to Authorization.

Authorization State. Event: The POP3 proxy server receives the **pass** command.

- If the Lock flag equals true, send response: -ERR Your mail box is locked by another POP3 session. The state remains as Authorization.
- If the password sent equals the password attribute in the user table entry and the time elapsed since the last-time-mailbox-empty time stamp is less than the global attenuation interval, set the terminate flag to true.

Additionally, set the Lock flag to true, start the Session Timer to wait for

possible timeout, and send a +OK response to the client. The state transitions to Transaction.

- If the password sent equals the password attribute in the user table entry and the time elapsed is greater than the global attenuation interval, set the Terminate flag to false. Additionally, send the **user** command to the POP3 mail server, wait for a response, send the **pass** command, and wait for a response. Return the response to the client. If the response from the server is -ERR, the state does not change. If the response is +OK, the state transitions to Transaction.
- If the password sent does not equal the password attribute in the user table entry, send the **user** command to the POP3 server, wait for a response, send the **pass** command, and wait for a response. Return the response to client. If the response from the server is -ERR, the state does not change. If the response is +OK, save the new password; the state transitions to Transaction.

Authorization State. Event: The POP3 proxy server receives the **quit** command. The Lock flag is set to false. The state transitions to idle.

Transaction State. Event: The POP3 proxy server receives any of these commands:

- **stat** -- If the terminate flag equals true, the POP3 proxy server returns +OK 0 0. This tells the client that their mailbox is empty. If the terminate flag equals false, the POP3 proxy server forwards the request to the POP3 mail server and returns the response to the client. The state remains as Transaction.

- 1           • **list** -- If the terminate flag equals true, the POP3 proxy server returns –ERR

2           no such message. If the terminate flag equals false, the POP3 proxy server

3           forwards the request to the POP3 mail server and returns the response to the

4           client. The state remains as Transaction.
- 5           • **retr** -- If the terminate flag equals true, the POP3 proxy server returns –

6           ERR no such message. If the terminate flag equals false, the POP3 proxy

7           server forwards the request to the POP3 mail server and returns the

8           response to the client. The state remains as Transaction.
- 9           • **dele** -- If the terminate flag equals true, the POP3 proxy server returns –

10          ERR no such message. If the terminate flag equals false, the POP3 proxy

11          server forwards the request to the POP3 mail server and returns the

12          response to the client. The state remains as Transaction.
- 13          • **noop** -- The POP3 proxy server returns +OK. The state remains as

14          Transition.
- 15          • **rset** -- The POP3 proxy server returns +OK. The state remains as

16          Transition.
- 17          • **top** -- If the terminate flag equals true, the POP3 proxy server returns –

18          ERR no such message. If the terminate flag equals false, the POP3 proxy

19          server forwards the request to the POP3 mail server and returns the

20          response to the client. The state remains as Transaction.
- 21          • **uidl** -- If the terminate flag equals true, the POP3 proxy server returns –

22          ERR no such message. If the terminate flag equals false, the POP3 proxy

23          server forwards the request to the POP3 mail server and returns the

24          response to the client. The state remains as Transaction.



- **quit** -- If the terminate flag equals true, the POP3 proxy server returns +OK. The Lock flag is set to false, and the state transitions to Idle. If the terminate flag equals false, the POP3 proxy server sends a stat command to the POP3 mail server, to retrieve the number of messages now in the mailbox. If the number is 0, the program sets the last-time-mailbox-empty time stamp to the current time; else it sets the time stamp to NULL. Then it sends quit to the POP3 mail server, and returns the result to the client. The Lock flag is set to false, and the state transitions to Idle.

Transaction State. Event: The POP3 Attenuation Session Timeout Event. The timer popped for a locally terminated session. The POP3 proxy server ends the session connection. The Lock flag is set to false, and the state transitions to Idle.

### **Garbage Collection**

The proxy server performs a garbage collection function. That is, at configurable intervals, or in case of need, it may delete all inactive entries in its user table. A case of need is defined, for example, as a user table that has become full. Other definitions are possible.

After the garbage collection, the proxy server can rebuild its user table, as new POP checks come in.

### **Opt Out List**

The proxy server may optionally support an Opt Out List. This is simply a list of user mailbox account names that won't be subject to proxy treatment. All POP3 commands for names on this list are passed through to the POP3 mail server.

### **The Fast Check Case**

CONFIDENTIAL

1           The proxy server may optionally support “fast check” logic to account for a case  
2       where a user knows that he has new mail (through some non-POP3 source of knowledge,  
3       as for example, when has just sent an e-mail to himself), and where he does rapid and  
4       repeated new mail checks within a short interval, to get the mail as soon as possible.

5           The basic function of the “fast check” logic is to keep track of the time of each new  
6       mail query of a user. If a certain number, say three checks, came through in a short  
7       interval, such as 30 seconds, the third check could be permitted through to the mail server.  
8       This would reward the user’s persistence. However, if the user continued to check rapidly  
9       after the third (where third is used as one example of a configurable value) check,  
10      subsequent checks would be terminated locally until another, longer, interval had expired.  
11      This would limit the effect of excessive, automated, or potentially malicious querying.

12          One way to discriminate the difference between fast checks that are the product of  
13      persistent manual checking by the user and automated checks where the check interval has  
14      been set unreasonably low (e.g., one every five seconds) is to assess the periodicity of the  
15      checks. If the time interval  $T$  between the quickly repeated checks is metronomically  
16      regular, then it is adjudged to be nothing more than automated checking. On the other  
17      hand, if the time interval  $T$  varies substantially, then it is adjudged to be manual fast  
18      checking. Some permissiveness is allowed for rewarding the persistence of a user who is  
19      fast checking manually. However, fast automated checks are attenuated ruthlessly.

20          Referring to **Fig. 2**, a logical implementation for a subset of the signaling  
21      transactions that take place between a client, a proxy server, and a mail server is  
22      illustrated. The signal transactions shown are appropriate for the case where the mail  
23      client does not already exist in the table.

1 Referring to **Fig. 3**, a logical implementation for another subset of the signaling  
2 transactions that take place between a client, a proxy server, and a mail server is  
3 illustrated. The signal transactions shown represent the case where the mail client is being  
4 attenuated (i.e., is in the table).

5 Referring to **Fig. 4**, a logical implementation for yet another subset of the signaling  
6 transactions that take place between a client, a proxy server, and a mail server is  
7 illustrated, according to an embodiment of the present invention. The signal transaction  
8 shown handles the situation where the client has changed their password.

9 Referring to **Fig. 5**, a logical implementation for still another subset of the  
10 signaling transactions that take place between a client, a proxy server, and a mail server is  
11 illustrated, according to an embodiment of the present invention. The signal transaction  
12 shown handles the situation where an erroneous password is used.

13 Referring to **Fig. 6**, a logical implementation for an additional subset of the  
14 signaling transactions that take place between a client, a proxy server, and a mail server is  
15 illustrated. The signal transaction illustrated implements synchronization for cache in  
16 transaction state (wherein messages have been cached).

17 Referring to **Fig. 7**, a logical implementation for a further subset of the signaling  
18 transactions that take place between a client, a proxy server, and a mail server according to  
19 an embodiment of the present invention. Possible synchronization for cache in transaction  
20 state (messages have been cached)

21 Preferably, the present invention is implemented according to soft state principles.  
22 Each proxy server should be able to build up its state from the information around it,  
23 without need for extensive backing up of previous state information. From an initial state  
24 of not knowing anything about user's or their accounts, the proxy server gradually learns

1 and builds a knowledge base, using only the algorithms it is programmed to implement  
2 and observation of the POP mail processes that occur around it.

3 For example, when a first POP check for an e-mail account is received by the  
4 proxy server (assuming it is starting from a no memory state), the proxy server permits the  
5 POP check to flow through to the mail server and starts to build a list. The proxy server  
6 determines the state of the mailbox and then keeps track of the state information. As this  
7 process repeats for different users, the proxy server builds up a table of users and the state  
8 of their respective mailboxes. This self-teaching aspect of soft state operation eliminates  
9 the need for expensive state back-up resources to store state information for later retrieval  
10 in the event of failure of the proxy server. A dead proxy server may simply be replaced by  
11 a similar machine what will teach itself what it needs to know.

12 The network traffic management goals of the above-described attenuation process  
13 may also be accomplished by delay of the POP checks. Instead of using the proxy server  
14 to report no new mail as disclosed above, the proxy server inserts a delay into the  
15 processing of the POP commands. As previously disclosed the first pop check results in  
16 the user being entered into a delay table. The user remains in the table for a time = N,  
17 where N is the desired allowable period between POP sessions. After time N the user will  
18 be dropped from the delay table.

19 This delay is preferably a variable that changes with client POP habits, or  
20 optionally the delay is a fixed value. Depending on the total amount of delay desired, this  
21 delay is inserted for a single POP command or for multiple of the POP session commands.  
22 The effect of this delay would be to increase the time that it takes for a POP session to  
23 complete.

1 While the delayed session is ongoing most email clients will not initiate a new  
2 session until the old session has been completed. If a new session is initiated, the protocol  
3 lock flag has been placed on the mailbox and an error will be sent -ERR Mailbox already  
4 open, for which the client will realize the last session is still open.

5 This delay should be less than the timeout setting in the POP3 client to avoid  
6 having the client report a timeout error. To allow for low settings of client timeout a small  
7 amount of delay can be added to multiple steps in the POP command. As mentioned  
8 above, the amount of delay can be set to a variable based on a user's POP check habits.  
9 That is, the amount of delay is programmable based on the frequency of POP checks. If  
10 the user is checking more frequently than desired for optimal network loading, the amount  
11 of delay is increased to keep the client from creating (or at least contributing to)  
12 overwhelming load. The delay is gradually decreased when the client reduces the  
13 frequencies of the POP checks. The quick check logic discussed above with regard to  
14 attenuation is also applicable to implementation of a POP delay scheme.

15 Referring to **Fig. 8**, a logical implementation for a subset of the signaling  
16 transactions that take place between a client, a proxy server, and a mail server is  
17 illustrated. The signal transactions shown represent the case where the mail client is being  
18 delayed (i.e., is in the delay table).

19 E-mail traffic load varies with time. There is little that can be done to control  
20 when e-mail messages are transmitted to a POP mail server. However, there is some  
21 opportunity to manipulate the traffic patterns for how and when e-mail is distributed from  
22 the mail server to the intended recipients. This may be exploited to minimize peak traffic  
23 loads to the extent that the load is caused by retrieval of e-mail messages.

1 According to the cache aspect of the invention, a user's e-mail is cached at the  
2 proxy server nearest to his presumed location. This decentralizes the e-mail storage away  
3 from the mail server and spreads it out over the network at the various proxy servers. This  
4 cache action is preferably done when there is a lull in network traffic (e.g., at night). This  
5 also has the effect of decentralizing the bandwidth demand on the overall network since  
6 the e-mail messages have a shorter distance to travel when retrieved by the user from the  
7 cache location at the proxy server.

8 By this cache action, the user's mail is moved out to the edges of the network  
9 ahead of when the user will be seeking to retrieve it. In a sense, the mail messages are  
10 pushed to the user, at least part of the way, to a proxy server near where it is presumed that  
11 the user will connect to the network when he seeks to retrieve his e-mail. This  
12 presumption will not always be accurate, however, since users are free to connect to the  
13 network at geographically diverse points (e.g., when they are traveling). More often than  
14 not, though, the presumption should be accurate.

15 One algorithm for choosing which proxy server to push a given user's e-mail to is  
16 to always assume that the same proxy server (entered in the user's profile as their "home"  
17 proxy server) will be appropriate. This algorithm does not adapt to a user's changing  
18 position. This algorithm relies only upon a default user profile as a reference, not a  
19 dynamic state table.

20 Another algorithm for choosing which proxy server to push a given user's e-mail  
21 to is to select the proxy server that is the most proximate to the last known point on the  
22 network where the user accessed the network. This algorithm is dynamic and adapts to a  
23 user's changing geographic position. However, this algorithm is not completely consistent  
24 with a soft state approach where there is no reliance upon keeping a memory of previous

1 state information. In the event of a loss of state information, the system may default to the  
2 algorithm described in the preceding paragraph.

3 In the event that the mail cache algorithm is wrong and sends email to the wrong  
4 proxy server to be cached, the user logging onto the network at another place (and  
5 interacting via an entirely different proxy server) may still retrieve the messages from the  
6 mail server. That is because the mail server does not delete the messages from its memory  
7 when it pushes the messages out to a selected proxy server. Until retrieved (from one  
8 server or another), the messages are redundantly stored on both servers simultaneously.

9 This redundant storage of e-mail messages raises issues of possible  
10 unsynchronized states. An unsynchronized state occurs when a user has read and deleted  
11 some messages from one server, but redundant copies of those messages remain stored on  
12 another server. This could result in a confusing situation where the user will retrieve a  
13 message that has already been retrieved and deleted in a previous e-mail session. To avoid  
14 such confusing occurrences, a synchronization algorithm may be used.

15 A synchronization algorithm according to one embodiment of the present invention  
16 makes use of unique identifiers (UIDs) that are assigned to each message. When e-mail  
17 messages are retrieved from a server (either from a proxy server or from the centralized  
18 mail server), a synchronization handshake occurs between the mail server and the relevant  
19 proxy server.

20 For example, if cached messages are retrieved from a proxy server, a  
21 synchronization inquiry is sent from the proxy server to the mail server inquiring whether  
22 the mail server contains messages in that user's mail box with the UIDs that match those  
23 of messages that the user has just retrieved locally. If there is identity of UIDs between  
24 the proxy server's cache mail box and the mail server's central mail box, then the user is

1 served the locally cached copies of the messages. When the user deletes a message, that  
2 message is deleted from both servers virtually simultaneously (allowing for transmission  
3 delay across the network). However, if there is a difference between the UIDs of  
4 messages residing on the two servers, the locally cached messages are thrown away and  
5 the messages stored at the mail server are retrieved for the user. In this manner, any  
6 confusion or conflict is avoided.

7 Referring to **Fig. 1**, a network diagram illustrates how proxy handling of e-mail  
8 may be implemented according to an embodiment of the present invention. An e-mail  
9 message originates from an originator client **20**, addressed to the e-mail account of an  
10 intended recipient client **40**. The e-mail message travels from the originator **20**, via a  
11 network **10**, to an e-mail server **30**. The process by which the e-mail message travels from  
12 the mail server **30** to the recipient **40** follows one of two general scenarios.

13 In the event that the recipient **40** is not operating their e-mail client software  
14 contemporaneously with the receipt of the e-mail message by the mail server **30**, the e-  
15 mail message may be cached. When traffic load on the network ebbs, the e-mail message  
16 is cached at a proxy server **50**, which is nominally local to the recipient **40**. When the  
17 recipient **40** subsequently initiates their e-mail client software, the recipient transmits a  
18 POP check (via the network **10**) that is intercepted by the proxy server **50**. In reply to the  
19 POP check, the proxy server **50** transmits to the recipient **40** the cached e-mail message  
20 and informs the e-mail server **30** that that message may be deleted from the e-mail server's  
21 memory.

22 In the event that the recipient **40** is operating its e-mail client software  
23 contemporaneously with the receipt of the e-mail message by the mail server **30**, the e-  
24 mail message is retrieved from the e-mail server **30** subject to any POP delay activity that



1 the proxy server 50 may perform. When the proxy server 50 permits a POP check from  
2 the recipient to pass through to the e-mail server 30, the e-mail server 30 transmits the e-  
3 mail message over the network 10 directly to the recipient 40.

4 The above description has made specific reference to POP protocols, however, the  
5 present invention is equally applicable to any mail protocol implemented in a network.

6 The present invention has been described in terms of preferred embodiments,  
7 however, it will be appreciated that various modifications and improvements may be made  
8 to the described embodiments without departing from the scope of the invention. The  
9 scope of the present invention is limited only by the appended claims.

WHAT IS CLAIMED IS:

1           1. A method of moderating traffic load on network servers in a network where  
2 electronic mail is retained for retrieval from at least one mail server, the method  
3 comprising:

4           permitting a mail request for a mail client to pass through a proxy server to the  
5 mail server; and

6           delaying subsequent mail requests for the mail client at the proxy server until a  
7 predetermined condition has been satisfied.

1           2. The method of claim 1, wherein the predetermined condition is a predetermined  
2 period of time.

1           3. The method of claim 2, wherein the predetermined period of time is  
2 dynamically determined based on the amount of traffic load on the network.

1           4. The method of claim 2, wherein the predetermined period of time is  
2 dynamically determined based on past behavior of the mail client.

1           5. The method of claim 4, further comprising:  
2 transmitting a message to the mail client providing information that the mail  
3 client's mail requests will be delayed as a result of the mail client's past behavior.

1           6. The method of claim 5, wherein transmitting a message comprises providing  
2 instructions on how to modify mail client software.



4 a memory including software instructions adapted to enable the proxy server to  
5 perform the steps of:  
6 permitting a mail request for a mail client to pass through the proxy server to  
7 the mail server; and  
8 delaying subsequent mail requests for the mail client at the proxy server until  
9 a predetermined condition has been satisfied.

1 15. The proxy server of claim 14, wherein the predetermined condition is a  
2 predetermined period of time.

1 16. The proxy server of claim 15, wherein the predetermined period of time is  
2 dynamically determined based on the amount of traffic load on the network.

1 17. The proxy server of claim 15, wherein the predetermined period of time is  
2 dynamically determined based on past behavior of the mail client.

1 18. The proxy server of claim 17, wherein the memory further includes software  
2 instructions to enable the proxy server to perform the step of:  
3 transmitting a message to the mail client providing information that the mail  
4 client's mail requests will be delayed as a result of the mail client's past behavior.

1 19. The proxy server of claim 14, wherein the predetermined condition is a  
2 combination of a predetermined time period and receipt of a notification from the mail  
3 server that mail has been received for the mail client at the mail server, whichever occurs  
4 first.



6 applying remedial handling of subsequent mail requests for the mail client at the  
7 proxy server until a predetermined condition has been satisfied, the remedial handling  
8 being selected from the group consisting of: delaying and attenuating; and  
9 transmitting a message to the mail client providing information that the mail  
10 client's mail requests will be delayed as a result of the mail client's past behavior  
11 wherein the predetermined condition is a predetermined period of time that is  
12 dynamically determined based on past behavior of the mail client.

## ABSTRACT OF THE DISCLOSURE

1 Traffic handling load on network servers is moderated by delaying POP checks at  
2 proxy servers located across the network. Delay of POP check processing is accomplished  
3 by intercepting each POP check packet at a proxy server that is nominally local to where  
4 the user is located. The proxy server permits a given user's initial POP check to proceed  
5 on through the network to the mail server. Thereafter, though, the proxy server only  
6 permits that user's received POP checks to proceed onward only after interjecting a delay  
7 in the POP process. Optionally, automatic messages are sent on a daily basis to a user  
8 who performs abusively rapid POP checks to inform them that their POP checks are being  
9 delayed and/or attenuated and educating them how to modify the configuration of their  
10 mail client software.

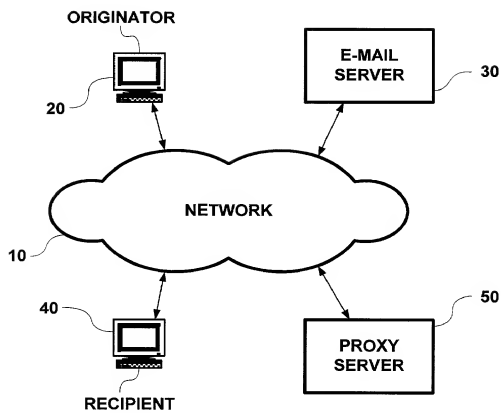


FIG. 1



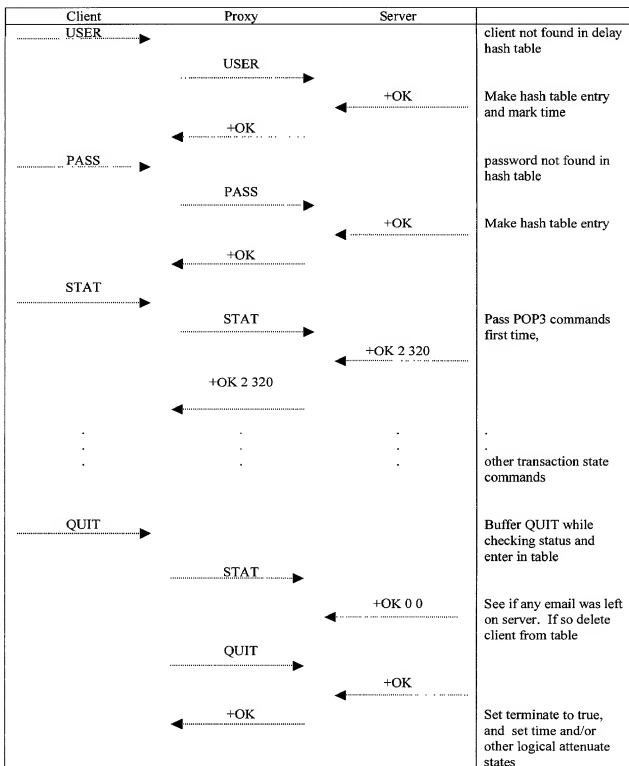


FIG. 2

002740: 80081560

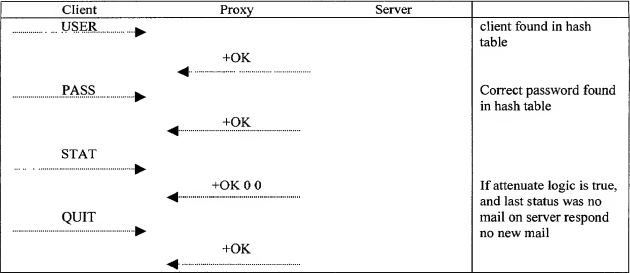


FIG. 3

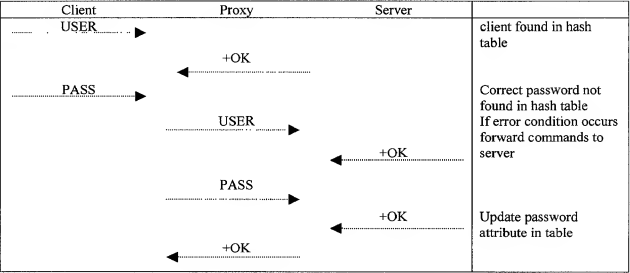


FIG. 4

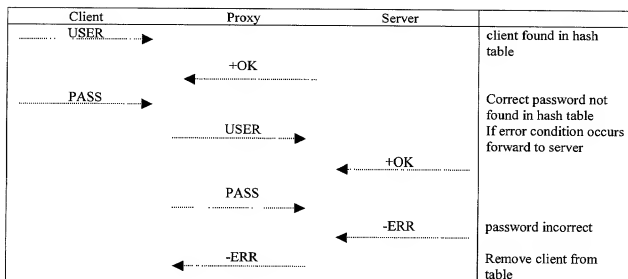


FIG. 5

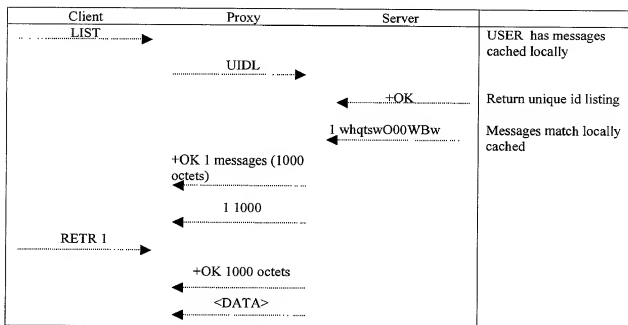


FIG. 6

00548306-041200

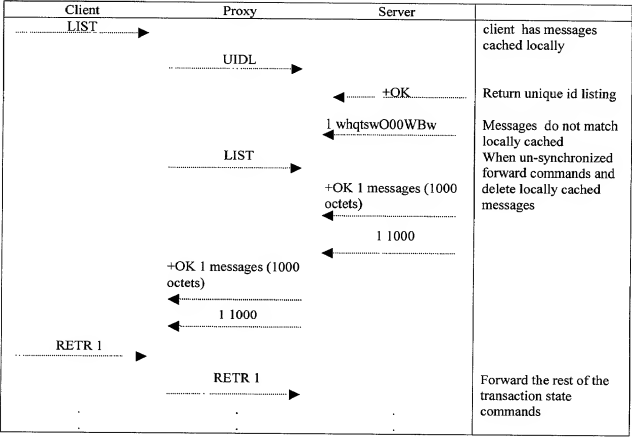


FIG. 7

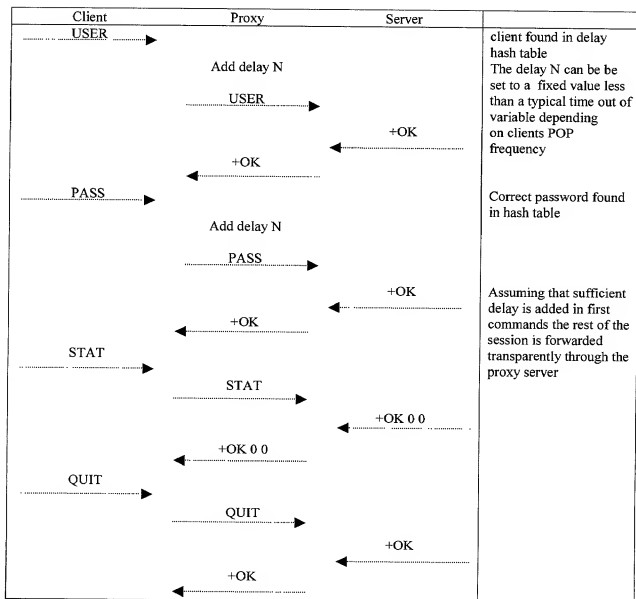


FIG. 8

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of Jeff Wasilko

Serial No.: Not Yet Assigned

Group Art Unit:

Filed: HEREWITH

Examiner:

**FOR: ATTENUATION, DELAY, QUEUING, AND MESSAGE CACHEING PROCESSES  
FOR USE IN E-MAIL PROTOCOLS IN ORDER TO REDUCE NETWORK SERVER  
LOADING**

**DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION**

As below inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, sole and first inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled **ATTENUATION, DELAY, QUEUING, AND MESSAGE CACHEING PROCESSES FOR USE IN E-MAIL PROTOCOLS IN ORDER TO REDUCE NETWORK SERVER LOADING**, the specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1.56(a).

I hereby appoint the following attorney(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Jon L. Roberts  
Registration No. 31,293  
John K. Abokhair, Esq.  
Registration No. 30,537  
Kevin L. Pontius  
Registration No. 37,512  
Christopher B. Kilner  
Registration No. 45,381  
Roberts Abokhair & Mardula, L.L.C.  
11800 Sunrise Valley Drive  
Suite 1000  
Reston, Virginia 20191-5302  
(703) 391-2900

002110.0000500

